```
Discussion 2020-07-29: String suggestion problem.
-------------------------------------------------

Minu
------
    >> Inverted Index
    >> Evaluation metric ??
    >> Correct, Correct word not predicted ??
    >> ?? if else expression ???

Binod
======
    >> Count vectorizer: Problem
    >> how to do it?

    (ngram generation completed)
    >> bigram
    >> tf-idf concept ??

Shankar
=======
    >> zero remove ??
    >> clojure variable ??
    >>

Astha
========
    >> Algorithm: inverted index, vectorizer
    >> Suggestion adequecy

Data generation
----------------

------------------
Removal of if-else expression
-----------------------------
Collection: you need to convert them.
[0, 1, 2, 6, 7] => apply isOdd

odd numbers

[] => ?

7 => [7] => isEmpty?, isNotEmpty?

filter, (fn[] ()), vector/list => final output.

-----------------------------------------------
Zero remove(default value)
Collection => getOrElse(key, defaultValue)
-----------------------------------------------

clojure variable
---------------------
const x => 100
reduce(Map()) {
    ..............
    (* x variable)
}
---------------------

Data generation
---------------
for()
```

```
first 100 prime numbers
-----------------------
>> infinite seq (lazy).fold((result [])).filter..... take(100)

-----------------------
tail recursion
--------------
f(10), 1
f(9),  10
f(8), 90
f(7), 720

...
f(1), result.
take result.

------------------------
for (i: 1 to 10) {
    i * result.
}

exapansion => first expand, and then evaluate.

generate, accumulate.

0, negative number.
--------------------
cond = [0, -1]

-907880347458972349875209384475908234745098273094857908283475098273402344598273405982323
9048523452345

count problem--
--------------
[a, b, c, a, b, c, a]
--------
group by
--------
[a (a, a, a), b (b, b), c (c, c)]
map (count values)=> [a 3, b 2, c 2]
-----------------------------------

Evaluation Metric
-----------------------------------
Job 1: Evaluation Data
----------------------
word list
10000

test words
100
w1      => w3, w790, w678, test data
w100    =>

-----------------------------------
w1      => system suggestion, w790, w3, w800

We can count hit and miss
-------------------------
hit: 2, miss: 1, insertion 1, deletion ?

hit => true position...
miss => false negative
insertion => false positive
```

```
deletion => ...

accuracy: correct result, incorrect result
correct => 1500
incorrect => 500
total => 2000
```